

Sala de Aula Invertida - Uma Experiência no Ensino-Aprendizagem de Programação para Administração de Redes de Computadores

Silmar Antonio Buchner de Oliveira – MPET/IFAM – silmarantonio@gmail.com

Andréa Pereira Mendonça – MPET/IFAM – andrea.mendonca@ifam.edu.br

Resumo: Neste artigo relatamos o planejamento e execução de uma proposta de ensino-aprendizagem de programação para administração de redes de computadores no modelo de sala de aula invertida, destinada a estudantes de Cursos Superiores de Tecnologia. A proposta pauta-se na resolução de problemas práticos, envolvendo automatização de soluções para monitoramento de equipamentos de redes e ações de usuários, análise de logs, sincronismos de dados e emissão automática de mensagens de alertas, implementados na linguagem de programação Python. Os resultados obtidos evidenciaram progresso dos estudantes, mas também a necessidade de ajustes na produção de conteúdos de nivelamento em conceitos básicos de redes e de programação. Toda a experiência com o modelo e os recursos utilizados serão compilados em um guia para utilização por outros professores.

Palavras chave: Ensino híbrido; Sala de aula invertida; Redes de computadores; Programação.

Flipped Classroom – An Experience in the Teaching-Learning of Programming for Computer Network Administration

Abstract: In this paper we present a teaching-learning proposal, following the flipped classroom model, of a course on programming for network administration aimed at students of undergraduate technological courses. This proposal is based on practical problem solving by using Python programming language and addressing automatization tasks such as for hardware and user actions monitoring, log analysis, data synchronization and automatic notifications. The students' progress was measured and revealed one of the positive outcomes of this course altogether with the need of levelling courses on fundamentals of computer networks and on programming. The whole experience with this teaching-learning proposal as well as the resources adopted are being synthesized in a guide that will be freely shared with other instructors.

Keywords: Blended-learning; Flipped classroom; Computer Networks; Programming.

1. Introdução

Considerado como uma das grandes tendências do século XXI, o ensino híbrido, do inglês *blended learning*, é uma abordagem que combina ensino presencial, em sala de aula, e *online*, no qual o estudante dispõe de tecnologias e algum controle sobre o tempo, lugar, modo e/ou ritmo do estudo (CHRISTENSEN; HORN; STAKER, 2013).

Há diferentes modelos de ensino híbrido relatados na literatura (HORN; STAKER, 2015), sendo um deles denominado sala de aula invertida (*flipped classroom*). Neste modelo, inverte-se a forma tradicional de ensino, isto é, o estudante passa a assistir a aula ou ter acesso ao conteúdo fora de sala, por meio de recursos como videoaulas, disponibilizadas previamente pelo professor. As tarefas práticas, tais como resolução de exercícios, que o estudante tradicionalmente executava em casa, passam a ser realizadas em sala de aula, com orientação do professor (BERGMANN; SAMS, 2012).

A implementação do modelo de sala de aula invertida propicia aos estudantes o desenvolvimento de autonomia e responsabilidade por sua própria aprendizagem, e eleva o papel do professor na medida em que este passa a auxiliar os estudantes na

aprendizagem mais profunda, orientando-os na resolução de problemas e esclarecimentos de dúvidas.

Estes benefícios nos motivaram a desenvolver uma proposta de ensino-aprendizagem de programação para administração de redes de computadores, destinada a estudantes de Cursos Superiores de Tecnologia (CST). A proposta foi implementada no formato de sala de aula invertida, no qual, fora de sala, os estudantes tinham acesso ao conteúdo por meio do Google Classroom e, em sala de aula, o tempo era dedicado à resolução de problemas práticos envolvendo monitoramento de equipamentos de redes e ações de usuários, análise de *logs*, sincronismos de dados e emissão automática de mensagens de alertas, sendo a programação desenvolvida na linguagem Python.

Os fundamentos desta proposta de ensino-aprendizagem, o percurso metodológico de sua elaboração, assim como os resultados obtidos com a sua implementação são apresentados nas próximas seções.

2. Sala de Aula Invertida e o Ensino de Redes de Computadores em CST

Nas próximas seções contextualizamos a dinâmica de implementação da sala de aula invertida, assim como detalhamos os desafios do ensino de programação para redes de computadores, considerando estudantes de Cursos Superiores de Tecnologia (CST), público alvo deste trabalho.

2.1 Sala de aula invertida

O modelo de sala de aula invertida foi idealizado por Bergmann e Sams (2012) no ano de 2006, quando lecionavam química em uma escola de ensino médio, relativamente rural, no interior do estado do Colorado (EUA). Por motivos particulares, não era incomum que os estudantes faltassem às aulas, o que acarretava retrabalho dos professores e perdas de conteúdos pelos estudantes. No intuito de sanar o problema, os dois professores resolveram gravar suas aulas em vídeos e postá-las *online*. Isto resolveu o problema dos faltosos e, colateralmente, beneficiou os outros estudantes, pois ainda que não faltassem, aderiram aos vídeos. Os estudantes relataram que isto melhorou seu aprendizado, pois poderiam voltar os vídeos aos pontos em que não haviam compreendido e rever o conceito sempre que necessário. Com isto, os professores resolveram gravar todas as suas aulas em vídeo (ao que chamaram de pré-aula) e propuseram que os estudantes os assistissem antes das aulas presenciais, que seriam destinadas à resolução de problemas envolvendo a teoria estudada. Com o sucesso da experiência, os professores resolveram aprimorar o modelo de ensino de sala de aula invertida e divulgá-lo para que outros professores pudessem aplicá-lo.

Segundo o modelo, a pré-aula concentra-se em vídeos construídos pelos professores ou cuidadosamente buscados *online*, de acordo com os conteúdos abordados. Durante a produção das aulas autorais é proposto que os professores busquem deixar seus vídeos intuitivos e atraentes, com uso de textos, anotações, inflexões de voz, etc. Os vídeos devem ser objetivos, não devendo ultrapassar quinze minutos de duração, onde cada tópico deve corresponder a um vídeo. (BERGMANN; SAMS, 2012).

É muito importante que os professores se preocupem com a adoção de tecnologias em benefício do ensino-aprendizagem, visto que esta geração de estudantes normalmente é bastante ambientada com vídeos, mensagens instantâneas e ambientes virtuais para diversos fins. Assim, diversos recursos podem ser utilizados na sala de aula invertida a fim de aproveitar esta afinidade, como Youtube, ambientes de aprendizagem como Moodle, Google Classroom, simuladores virtuais de máquinas e ambiente

computacionais como VirtualBox ou VMWare¹ e também comunicação via Whatsapp, Telegram e *e-mails*.

Ao contrário do que possam parecer, os recursos tecnológicos adotados antes da aula presencial não são o maior benefício da sala de aula invertida, mas sim, o tempo liberado para atividades práticas presenciais com maior qualidade e mais envolvimento dos estudantes em estratégias auxiliares como a aprendizagem baseada em projetos e/ou problemas. Além disso, com mais tempo livre, o professor pode auxiliar melhor a aplicação do conteúdo, tirar dúvidas e direcionar maior atenção aos estudantes com dificuldades nas aulas presenciais (BERGMANN; SAMS, 2012).

Tais características revelam-se benéficas ao contexto do ensino-aprendizagem de programação para redes de computadores, conforme descrevemos a seguir.

2.2 Ensino de redes de computadores em CST

Os Cursos Superiores de Tecnologia (CST) são uma modalidade de ensino superior com duração mais curta (entre 2 e 3 anos), se comparado a cursos de Bacharelado e, geralmente, são oferecidos pelos Institutos Federais de Educação, pertencentes à rede Federal de Educação Profissional, Científica e Tecnológica, criada em 2008. Há também a oferta de CST em universidades públicas e Instituições de Ensino Superior particulares (BRASIL, 2006; BRASIL, 2002).

O Catálogo Nacional de Cursos Superiores de Tecnologia organiza e orienta a oferta de cursos superiores de tecnologia, inspirado nas diretrizes curriculares nacionais e em sintonia com a dinâmica do setor produtivo e as expectativas da sociedade (SETEC/MEC). O Catálogo é organizado por eixos, sendo o eixo Informação e Comunicação aquele que integra os cursos relacionados à área de Computação/Informática, como CST em Análise e Desenvolvimento de Sistemas, CST em Sistemas para Internet, CST em Redes de Computadores, CST em Jogos digitais, entre outros (BRASIL, 2016).

Independente da denominação, os Cursos do eixo Informação e Comunicação possuem em seus currículos, pelo menos, uma disciplina de Redes de Computadores. De acordo com uma pesquisa *ad-hoc* que realizamos em sessenta (60) CST a partir de informações disponibilizadas nos *sites* de quarenta (40) Institutos Federais de Educação, observamos que as ementas são baseadas essencialmente em teorias sobre pilhas de protocolos TCP/IP e no modelo de referência OSI/ISO, não havendo menções a aplicações práticas nas disciplinas iniciais.

Em contrapartida a este cenário de formação, o mercado de trabalho requer dos egressos dos CST a resolução de problemas práticos e, no caso da área de redes de computadores, são comuns demandas voltadas para administração de redes de computadores, as quais envolvem automatização de procedimentos para monitoramento, controle e gerenciamento de serviços, como execução de *backups*, sincronismo de dados, verificação de consumo de recursos, envio de mensagens de alerta via *e-mail*, etc.

Tendo em vista a melhoria da formação destes estudantes, concebemos uma proposta de ensino-aprendizagem de programação para administração de redes de computadores, sendo possível implementá-la desde as disciplinas iniciais de redes de computadores. A concepção desta proposta encontra sustentação em diferentes autores, dentre eles, Kurose et al. (2002) que enfatizam que estudantes de redes de computadores que aprendem a codificar programas podem ter mais facilidade para internalizar e entender conceitos avançados, além de deixá-los com um conhecimento “comercializável”.

¹ VirtualBox e VMWare são *softwares* para virtualização de máquinas e sistemas operacionais.

Para que a proposta de ensino-aprendizagem oportunizasse aos alunos maior experiência “mão na massa”, adotamos o modelo de sala de aula invertida, combinando estudo autônomo, para apropriação dos conceitos, com estudo presencial, no qual o tempo de sala de aula era dedicado a resolução de problemas práticos, conforme metodologia descrita a seguir.

3 Metodologia

Nesta seção ressaltamos a metodologia de concepção da proposta de ensino-aprendizagem que demandou seis meses de trabalho, incluindo planejamento de conteúdos, organização dos estudos presenciais e autônomos, seleção de tecnologias e produção de videoaulas.

3.1 Organização do conteúdo

Os assuntos foram selecionados a partir de uma busca por livros nacionais e estrangeiros que envolvessem administração de redes de computadores. Dos trinta livros iniciais pesquisados, selecionamos dezesseis e, neste subconjunto, selecionamos os assuntos mais frequentes e identificamos também os tipos de problemas associados aos conteúdos, conforme ilustrado no Quadro 1.

Quadro 1 - Relação de assuntos *versus* ocorrência no conjunto de livros.

Assunto	Ocorrências
Monitoramento de Redes	9
Compartilhamento de dados	8
Envio de <i>e-mails</i> /relatório	8
Gerenciamento de sistemas de arquivos	8
Gerenciamento de usuários	8
<i>Login</i> (acesso de usuários)	7
Monitoramento de sistemas	7
Gerenciamento de Arquivos e diretórios	7
<i>Backup</i>	6
SSH/acesso remoto	6
Análise de <i>logs</i> (registros de atividades)	4
Compressão de dados	4
Expressões regulares	3
Manipulação de <i>strings</i>	3

Fonte: Autoria própria.

Organizamos os assuntos por temáticas e, para uma melhor estruturação, dividimos a proposta de ensino-aprendizagem em dois módulos: o módulo 1 para tratar sobre monitoramento de redes e de ações de usuários; e, o módulo 2 para tratar sobre preservação das informações e comunicação de eventos. Em cada módulo foram abordadas 3 temáticas, cada uma englobando um ou mais assuntos. O grau de complexidade dos assuntos, questões e problemas aumentava ao longo de cada módulo conforme evoluíam as temáticas, que, por sua vez, integravam conteúdos sobre redes de computadores, sistemas operacionais e programação, conforme pode ser observado no Quadro 2.

Seguindo os preceitos da sala de aula invertida, proposto por Bergmann e Sams (2012), cada temática foi tratada seguindo duas abordagens. A primeira chamamos de “estudo autônomo”, na qual o estudante, fora da sala de aula, apropria-se dos conceitos envolvidos; e a segunda, chamamos de “estudo presencial”, que ocorre com a presença do professor em um laboratório de informática, a fim de dirimir dúvidas e solucionar problemas práticos, articulando os conceitos aprendidos durante o estudo autônomo.

Quadro 2 – Módulos, temas, assuntos, conteúdos e problemas.

Módulo 1 - Monitoramento de redes de computadores		
<p>Tema 01: Identificando a movimentação da vizinhança Assunto: Monitoramento de disponibilidade de equipamentos e serviços em rede. Conteúdos principais: <u>Redes:</u> Comando “ping”; Protocolo SNMP. <u>Programação:</u> Bibliotecas e métodos; variáveis; E/S de dados, <u>S.O. Linux:</u> Estruturas de arquivos e diretórios; Monitoramento de estados de componentes. Problemas propostos: Detecção de problemas de comunicação, através de monitoramentos via SNMP e “ping”.</p>	<p>Tema 02: Cuidando do <i>Habitat</i> Assunto: Monitoramento de estado de discos e partições, em rede. Conteúdos principais: <u>Redes:</u> Protocolo NFS; Protocolo SNMP. <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados, <u>S.O. Linux:</u> Compartilhamento de dados; Estruturas de arquivos e diretórios; Problemas propostos: Monitoramentos remotos de discos e partições como prevenção de estouros de espaços de armazenamento.</p>	<p>Tema 03: De olho nas intenções Assunto: Monitoramento de ações de usuários. Conteúdos principais: <u>Redes:</u> Protocolo SSH, Acesso remoto; <u>Programação:</u> Bibliotecas e métodos; variáveis; E/S de dados; Arquivos; <i>Strings</i> e datas; Laços; Decisão, <u>S.O. Linux:</u> Editores de textos; Acesso remoto; Arquivos de configuração; Históricos de ações; Data e hora; Problemas propostos: Monitoramentos de ações de usuários através análise <i>logs</i> do sistema operacional.</p>
Módulo 2 - Preservação de informações e comunicação de eventos		
<p>Tema 04: Construir é difícil. Reconstruir pode ser impossível Assunto: Compactação de dados, <i>backup</i> em rede e registros de atividades (<i>logs</i>). Conteúdos principais: <u>Redes:</u> <i>Backup</i>; Compartilhamento de arquivos; Protocolo NTP; Protocolo NFS. <u>Programação:</u> Bibliotecas e métodos; variáveis; E/S de dados; <i>Strings</i> e datas; Laços; Decisão. <u>S.O. Linux:</u> Empacotamento e compactação de dados; Agendamento de tarefas; Sincronismo de data e hora; Arquivos de <i>logs</i>. Problemas propostos: Agendamentos <i>backups</i> em rede, com verificação de etapas de execução e gravação de <i>logs</i>.</p>	<p>Tema 05: Um é pouco. Então tenha dois (ou mais!) Assunto: Sincronismo de dados em rede. Conteúdos principais: <u>Redes:</u> <i>Backup</i>; Políticas de <i>backup</i>; Sincronismo de dados remotos; Compartilhamento de arquivos; Protocolo SSH; Protocolos NFS. <u>Programação:</u> Bibliotecas e métodos; variáveis; E/S de dados; <i>Strings</i> e datas; Decisão. <u>S.O. Linux:</u> Sincronismo de dados; Agendamento de tarefas; Arquivos de <i>logs</i>. Problemas propostos: Agendamentos de sincronismo de dados em rede, com verificação de etapas de execução e gravação de <i>logs</i>.</p>	<p>Tema 06: Seja o primeiro a saber Assunto: Automatização de mensagens de eventos e de alertas via <i>e-mail</i> e mensageria instantânea (Telegram). Conteúdos principais: <u>Redes:</u> Serviços de <i>e-mail</i>; Protocolo SMTP; Protocolos SSL/TLS e StartTLS. <u>Programação:</u> Bibliotecas e métodos; variáveis; E/S de dados; <i>E-mails</i>; <i>Strings</i>. <u>S.O. Linux:</u> <i>E-mails</i>; mensageria instantânea; Sincronismo de dados; Agendamento de tarefas. Problemas propostos: Automatizações de procedimentos com verificações de etapas de execuções e envio de alertas via <i>e-mail</i> e Telegram.</p>

Fonte: Autoria própria.

3.2 Elaboração dos Roteiros de Aprendizagem para estudo Autônomo e Presencial

Os Roteiros de Aprendizagem (RA) funcionam como guias para que o estudante alcance os resultados pretendidos da aprendizagem em seus estudos autônomos e presenciais. Para cada temática foram criados dois RA, um para o estudo autônomo e outro para o estudo presencial, totalizando doze (12) RA. Os RA para estudo autônomo e presencial são semelhantes em sua estrutura geral, diferenciando-se na seção de tarefas. Nos estudos autônomos, as tarefas são compostas por questões envolvendo conceitos e também pequenas aplicações práticas em um laboratório virtual, criado especificamente para dar suporte à proposta de ensino-aprendizagem. Para apoiar os estudantes na resolução das tarefas são indicados, nos próprios RA, vídeos, tutoriais *online* e *links* de páginas *web*.

Nos RA para estudo presencial, as tarefas foram inspiradas em Bender (2014) e em Biggs e Tang (2011) e são constituídas da seguinte dinâmica: (i) *aquecimento*: um tempo é destinado para que os estudantes exponham suas dúvidas sobre o estudo autônomo, em uma discussão mediada pelo professor; (ii) *apresentação do contexto*: é apresentado um contexto envolvendo um problema real na área de redes de computadores; (iii) *discussão*: são apresentadas questões que instigam o estudante a compreender melhor o problema e encontrar meios para solucioná-lo; (iv) *resolução*: é

solicitado dos estudantes a implementação de um programa, em linguagem Python, para solucionar o problema proposto, executando-o no laboratório virtual.

3.3 Laboratório virtual

Presencialmente, no laboratório de informática o estudante encontrava um ambiente virtual de redes para testar seus programas. Contudo, para que pudesse cumprir com as tarefas do RA autônomo, foi necessária a reprodução do ambiente virtual em seus computadores pessoais.

Para auxiliar os estudantes, elaboramos um tutorial para os ajudar a reproduzir o laboratório virtual que continha 3 servidores Linux. Um estudo prévio foi feito para que este laboratório pudesse ser facilmente replicado pelo estudante em seu computador, de forma que fosse executado sem consumir muitos recursos como memória e processamento. Desta forma, optamos pelo Ubuntu Minimal por ser uma versão básica do Linux² que possibilitava facilmente a instalação de quaisquer funcionalidades posteriores.

Para facilitar a reprodução dos servidores, arquivos automatizados de configuração foram disponibilizados em uma máquina virtual “base” que o estudante baixaria e importaria, no seu computador pessoal, para a plataforma de virtualização Oracle VM Virtualbox³. A partir desta importação, o estudante executaria a clonagem da máquina base, obtendo os três servidores. Um destes servidores possuía interface gráfica e tinha instalado o ambiente Thonny⁴ para desenvolver suas soluções na linguagem Python.

3.4 Ambiente Virtual e Recursos Digitais para guiar o estudo Autônomo e Presencial

O Google Classroom (no Brasil, Google Sala de Aula) é uma plataforma *web* da Google que reúne um conjunto de serviços (*e-mail*, *drive*, gerenciador de tarefas, entre outros) e foi utilizado para disponibilizar os materiais didáticos (RA, videoaulas, tutoriais, apostilas, páginas *web*, etc.) e promover as interações virtuais, também feitas por Whatsapp.

Para dar suporte ao estudo autônomo, produzimos vídeos autorais e também utilizamos vídeos de terceiros, especialmente selecionados para apoiar o estudo das temáticas. Cada vídeo tinha em média sete (07) minutos e foram produzidas 27 videoaulas autorais (totalizando, 3 horas e 28 minutos) e selecionados 17 vídeos de terceiros (totalizando, 1 hora e 48 minutos). A produção e a seleção dos vídeos foram realizadas após a determinação dos temas e conteúdos. No caso das videoaulas autorais, antes de cada gravação, um roteiro foi escrito para que o vídeo fosse objetivo e relativamente curto. Para gravação, inserção de textos explicativos, anotações, edição e acréscimo de efeitos foi utilizado o *software* Camtasia Studio por agregar todas estas funções. A publicação dos vídeos autorais foi feita no Youtube. Além dos vídeos, foram disponibilizados outros recursos como tutoriais *online*, conteúdos de páginas *web* e indicação de leitura de capítulos de livros.

² O Linux é um sistema operacional bastante difundido, executado em mais de dois terços dos servidores *web*. Também possibilita facilmente outros serviços de rede, além de ser *software* livre (com código aberto, gratuito), com vasto material *online* e comunidades ativas. A escolha pelo Linux Ubuntu se deveu a sua popularidade, pois aproximadamente metade dos servidores Linux identificados na Internet utilizam esta distribuição (NETCRAFT, 2017).

³ A plataforma de virtualização Oracle VM Virtualbox foi escolhida por ser um *software* livre e dotado de interface intuitiva.

⁴ A IDE Thonny é descrita pelos seus idealizadores como “IDE para iniciantes”. Foi escolhida por ter uma interface amigável, por ser “leve” e multiplataforma (THONNY.ORG, 2018).

3.5 Avaliações

Planejamos a realização de três avaliações (uma diagnóstica e duas modulares), aplicadas individual e presencialmente. A avaliação diagnóstica foi estruturada em duas partes: uma, abordando teoria, na forma de questões abertas e de múltipla escolha; e outra abordando problemas práticos envolvendo a administração de redes, na qual os estudantes deveriam prover soluções na forma de programas. A avaliação diagnóstica foi realizada no primeiro encontro e tinha por objetivo capturar os conhecimentos prévios dos alunos, tanto do ponto de vista conceitual, quanto de programação.

As avaliações modulares foram aplicadas ao final de cada módulo e compreendiam apenas problemas práticos de redes, para os quais os alunos deveriam implementar programas para solucioná-los. Estas avaliações tinham por objetivo prover *feedback* sobre quão bem os estudantes alcançaram os resultados pretendidos da aprendizagem, que diziam respeito a automatização de soluções para problemas envolvendo administração de redes de computadores.

4 Implementação da Proposta

A proposta de ensino-aprendizagem foi implementada na forma de um curso de extensão, com carga horária de 60 (sessenta) horas, sendo 40 (quarenta) horas para estudos presenciais e 20 (vinte) horas para estudos autônomos. O curso foi oferecido pelo Instituto Federal de Rondônia, Campus Zona Norte, para estudantes de CST na área de Informática, sendo apenas exigido como pré-requisito o estudante ter cursado previamente alguma disciplina de programação. O curso foi realizado entre novembro de 2017 e fevereiro de 2018 e contou com a participação efetiva de 15 estudantes que concluíram o curso.

Ressaltamos que está é uma pesquisa de natureza aplicada, com abordagem predominantemente qualitativa, pois levou em conta a observação dos fenômenos durante a aplicação da proposta de ensino.

No primeiro dia do curso a proposta de ensino foi apresentada aos estudantes, assim como a dinâmica dos estudos por meio da sala de aula invertida. Neste dia também foi realizada a avaliação diagnóstica. No geral, o desenvolvimento do curso ocorreu da seguinte maneira: a cada início de semana, o professor disponibilizava o RA autônomo e os materiais didáticos no Google Classroom, ficando disponível para tirar dúvidas por *e-mail* e, principalmente, por Whatsapp em um grupo criado para este fim. No sábado pela manhã, facultativamente, em um horário das 9:00 às 11:30 havia uma tutoria com a presença do professor em que os estudantes poderiam elucidar suas dúvidas sobre os estudos autônomos e utilizar o laboratório de informática.

Nos estudos presenciais, aos sábados, das 14:00 às 18:00, os estudantes tinham acesso ao RA para estudo presencial, que os guiaria na resolução de problemas práticos. Com o conteúdo básico previamente trabalhado pelos estudantes, ao professor cabia a orientação da parte prática e elucidação de dúvidas durante o percurso de resolução de problemas, com oportunidade de atender melhor às dificuldades individuais. Após o estudo presencial, as tarefas eram corrigidas e, durante a semana, um *feedback* sobre o desempenho era dado a cada estudante, contendo apontamentos sobre quais conteúdos ou habilidades deveria dedicar mais atenção.

As avaliações modulares, nas datas marcadas, eram disponibilizadas em formulários eletrônicos no Google Classroom, realizadas individualmente pelos estudantes no laboratório de informática, com o uso de computadores e do laboratório virtual. As questões destas avaliações requeriam programação em Python e as soluções eram desenvolvidas na IDE Thonny, no ambiente Linux e as respostas, bem como os códigos resultantes, eram copiados no formulário eletrônico.

5 Resultados

Dos estudantes que participaram do curso, 80% eram de estudantes do sexo masculino e 20% do sexo feminino e a média de idade era de 24,2 anos. Do total, 60% dos alunos cursavam o segundo período de seus respectivos cursos, 27% cursavam o quarto período e 13% cursavam o sétimo ou oitavo período. Quanto aos cursos superiores, 53% estavam matriculados e cursando CST em Redes de Computadores, 20% CST em Análise e Desenvolvimento de Sistemas, 20% Bacharelado em Sistemas de Informação e 7% Engenharia da Computação. Todos informaram terem cursado disciplinas de redes de computadores e de programação em seus respectivos cursos superiores.

Os resultados da avaliação diagnóstica revelaram dificuldades dos estudantes com a resolução de problemas envolvendo programação, pois apenas seis estudantes conseguiram desenvolver alguma solução, ainda que seus aproveitamentos individuais não tenham ultrapassado 25% de acertos. Também foram evidenciadas dificuldades com questões básicas sobre teorias de redes de computadores, pois mais da metade dos estudantes erraram questões objetivas envolvendo protocolos ICMP (*Internet Control Message Protocol*) e o comando PING; SNMP (*Simple Network Management protocol*) e nomenclaturas básicas; protocolos de compartilhamento de arquivos e diretórios; e noções básicas sobre TCP/IP e OSI/ISO. Havia uma expectativa maior sobre este desempenho, visto que 73,3% dos estudantes assumiram ter maior afinidade com redes de computadores do que com programação e que todos haviam cursado ao menos uma disciplina de redes de computadores.

Nos estudos autônomos, observamos dificuldades dos estudantes para estudarem sozinhos, pois era uma prática nova para a maioria. Do total de alunos, apenas 33,3% deles afirmaram estudar mais de 2 horas por semana fora da sala de aula. Essa dificuldade se refletia nos estudos presenciais, havendo ocasiões em que a fase de aquecimento teve de ser estendida de 30 minutos para 1 hora, atrasando o início das tarefas práticas. Para ajudar os estudantes a mitigar esta dificuldade, conversamos com o grupo sobre estratégias para melhorar a gestão do tempo e a importância do estudo além do espaço de sala de aula. No decorrer do curso o início das atividades práticas nos estudos presenciais foi se normalizando e, nos últimos temas a dedicação para os estudos autônomos havia aumentado para 3 horas semanais, em média.

Foram entregues concluídos, em média, 86% dos RA para estudos autônomos e 87% dos RA para estudos presenciais, ainda que alguns estudantes tenham atrasado suas entregas. As alegações para os atrasos envolviam problemas pessoais e conflitos com as atividades dos estudantes em seus cursos regulares.

Para avaliarmos os recursos disponibilizados, ao final de cada encontro presencial os estudantes eram questionados sobre a qualidade dos vídeos e de outros materiais disponibilizados. De forma geral, os estudantes avaliaram estes recursos como adequados ao tempo e ao tema de cada estudo. Houve solicitação de acréscimo de materiais sobre Linux e redes de computadores. Em virtude disso, produzimos uma apostila básica sobre os assuntos e dedicamos quatro horas para uma aula extra presencial antes do final do primeiro módulo, o que resultou em uma queda acentuada quanto às dúvidas sobre tais assuntos no decorrer do curso.

No início do curso, 55,87% dos estudantes afirmaram recorrer a outras fontes além das indicadas nos RA autônomos, principalmente sobre conceitos de programação em Python, ainda que tivéssemos indicado vídeos sobre todos os conteúdos de programação exigidos no curso. Fizemos um trabalho individual incentivando uma melhor exploração do material e ao final do curso apenas 15,38% afirmaram ter de recorrer a materiais extras em seus estudos autônomos.

Para aferirmos a evolução dos estudantes quanto as suas habilidades para resolução de problemas de redes com recursos de programação, levamos em conta, além das avaliações, os estudos presenciais. Notamos dificuldades ao longo das três primeiras temáticas e atuamos junto a cada estudante, presencial e remotamente, para elucidar suas dúvidas. Na avaliação intermediária, ocorrida ao final do primeiro módulo, a média da turma foi 6,8 pontos (em uma escala de 0 a 10), acima da nota mínima para aprovação no curso e também acima da média aceita para aprovação nos cursos superiores regulares que os estudantes cursavam, ambos de 6,0 pontos. Na avaliação final, realizada após o término do segundo módulo, a média da turma foi de 8,7 pontos (em uma escala de 0 a 10), representando uma melhora significativa no desempenho do grupo.

Acreditamos que a implementação desta proposta de ensino como uma disciplina do currículo pode contribuir para um melhor desempenho dos estudantes. Na forma de um curso de extensão, demandou atividades extras para além das atividades desenvolvidas pelos alunos no período letivo, principalmente se considerarmos a natureza aplicada da proposta e a necessidade de estudo autônomo. Este é um aspecto que será mais bem avaliado em trabalhos futuros.

Com respeito à adoção do Google Classroom, observamos ampla aceitação por parte dos estudantes, principalmente para o compartilhamento de materiais. Contudo, esta plataforma não foi muito utilizada pelos alunos para as interações virtuais, neste caso, os alunos preferiram utilizar o Whatsapp.

6 Considerações Finais

Neste trabalho evidenciamos a trajetória de preparação e aplicação de uma experiência de ensino-aprendizagem pautada no modelo de sala de aula invertida para desenvolver em estudantes de nível superior conhecimento para a resolução de problemas práticos de redes de computadores, com recursos de programação em Python.

Em outros trabalhos encontramos relatos de aplicações de sala de aula invertida como em Santos et al. (2017), que relataram resultados positivos com a aplicação do modelo no ensino médio. Chen (2014) descreveu melhoria nas notas finais dos estudantes após aplicação do modelo no ensino de programação em C em curso superior. Oliveira et al. (2017) propuseram uma série de análises sobre a aplicação de metodologias semelhantes em disciplinas de programação em C. Medeiros e Bessa (2017) propuseram uma ferramenta para análise de avaliações em grandes turmas, utilizando o modelo de sala de aula invertida. Leite, Hoji e Abdalla Junior (2018) descrevem os desafios para a implementação do modelo de sala de aula invertida em disciplinas de Redes de Computadores em um curso técnico. Diferentemente destes trabalhos, além de implementar o modelo de sala de aula invertida, temos por objetivo conceber um guia didático contendo o planejamento, a metodologia detalhada da implementação, assim como os recursos empregados, para que a experiência possa ser reproduzida por outros professores.

Como trabalhos futuros, pretendemos também melhorar alguns aspectos da proposta, como a adição de videoaulas para nivelamento sobre redes de computadores, programação em Python e Linux. Adicionalmente, pretendemos implementar a proposta em uma disciplina regular de redes de computadores.

Referências

BENDER, W. N. **Aprendizagem baseada em projetos** – Educação diferenciada para o Século XXI. Porto Alegre: Penso, 2014.

BERGMANN, J; SAMS, A. **Flip YOUR Classroom** – Reach Every Student in Every Class Every Day. Alexandria: ISTE, 2012.

BIGGS, J.; TANG, C. **Teaching for Quality Learning at University**. 4. ed. Berkshire, England: Society for Research into Higher Education & Open University Press, 2011.

BRASIL. Ministério da Educação (2002). **Parecer Conselho Nacional de Educação** – Conselho Pleno nº 29/2002. Institui as Diretrizes Curriculares Nacionais Gerais para a organização e o funcionamento dos cursos superiores de tecnologia. Disponível em: <<http://portal.mec.gov.br/cne/arquivos/pdf/cp29.pdf>>. Acesso em: 28 abr. 2018.

BRASIL. Ministério da Educação (2006). **Decreto n.º 5.773, de 9 de maio de 2006**. Dispõe sobre o exercício das funções de regulação, supervisão e avaliação de instituições de educação superior e cursos superiores de graduação e sequenciais no sistema federal de ensino. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2006/decreto/d5773.htm>. Acesso em: 28 abr. 2018.

BRASIL. Ministério da Educação. **Catálogo Nacional de Cursos Superiores de Tecnologia**. 3 ed. Brasília: jun. 2016. Disponível em: <http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=44501-cncst-2016-3edc-pdf&category_slug=junho-2016-pdf&Itemid=30192>. Acesso em: 28 abr. 2018.

CHEN, E. Z., Flipped Classroom Model and Its Implementation in a Computer Programming Course. **Lärlärdom**, vol. 2014, p. 180-200, 2014. Disponível em: <<http://www.diva-portal.org/smash/get/diva2:777869/FULLTEXT01.pdf>>. Acesso em: 28 abr. 2018.

CHRISTENSEN, C.; HORN, M.; STAKER, H. Ensino Híbrido: uma Inovação Disruptiva? Uma introdução à teoria dos híbridos. **Clayton Christensen Institute for disruptive innovation**, 2013. Disponível em: <https://s3.amazonaws.com/porvir/wp-content/uploads/2014/08/PT_Is-K-12-blended-learning-disruptive-Final.pdf>. Acesso em: 28 abr. 2018.

HORN, M.; STAKER, **Blended - Using Disruptive Innovation to Improve Schools**. San Francisco: Jossey-Bass, 2015.

KUROSE, J. F.; et al. Workshop on computer networking: curriculum designs and educational challenges. **ACM SIGCOMM Computer Communications Review**, v. 32, n. 5, p. 1–9, Nov. 2002. Disponível em <Workshop on computer networking: curriculum designs and educational challenges>. Acesso em: 28 abr. 2018.

LEITE, F. N.; HOJI, E. S.; ABDALA JUNIOR, H., A Blended Learning Method Applied in Data Communication and Computer Networks Subject. **IEEE Latin America Transactions**, v. 16, n. 1, jan. 2018. p. 163-171. Disponível em: <http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol16/vol16issue01Jan.2018/16TLA1_25NLeite.pdf>. Acesso em: 28 abr. 2018.

MEDEIROS, R. A. C; BESSA, A. MiniTeste: uma ferramenta ágil para aplicação de avaliações personalizadas. **RENOTE - Revista Novas Tecnologias na Educação**, Porto Alegre, v. 15, n. 1, jul. 2017. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/75126>>. Acesso em: 13 mai. 2018.



NETCRAFT. **September 2017 Web Server Survey** - Web-facing operating systems. Disponível em: <<https://news.netcraft.com/archives/2017/09/11/september-2017-web-server-survey.html>>. Acesso em: 28 abr. 2018.

OLIVEIRA, M. G. et al. Um Curso de Programação a Distância com Metodologias Ativas e Análise de Aprendizagem por Métricas de Software. **RENOTE** - Revista Novas Tecnologias na Educação, Porto Alegre, v. 15, n. 1, jul. 2017. Disponível em: <<http://www.seer.ufrgs.br/renote/article/view/75143>>. Acesso em: 13 mai. 2018.

SANTOS, A. C. et al. Ensino Híbrido: Relato de Experiência sobre o uso de AVEA em uma proposta de Sala de Aula Invertida para o Ensino Médio. **RENOTE** - Revista Novas Tecnologias na Educação, Porto Alegre, v. 15, n. 2, dez. 2017. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/79186/46020>>. Acesso em: 28 abr. 2018.

THONNY.COM. **Python IDE for beginners**. Disponível em <<http://thonny.org>>. Acesso em: 28 abr. 2018.